

COPY OF PAPERS
ORIGINALLY FILED

#4

U.S. DEPARTMENT OF COMMERCE
PATENT AND TRADEMARK OFFICE

**CLAIM TO CONVENTION PRIORITY
UNDER 35 U.S.C. § 119**

Docket Number:
10191/2274

Application Number
10/092,225

Filing Date
March 5, 2002

Examiner
To be assigned

Art Unit
2121

Invention Title
**METHOD AND DEVICE FOR DETERMINING
THE LOAD OF A COMPUTING ELEMENT**

Inventor(s)
Jens BOETTCHER

Address to:
Assistant Commissioner for Patents
Washington D.C. 20231

I hereby certify that this correspondence is being deposited with the
United States Postal Service as first class mail in an envelope addressed
to: Assistant Commissioner for Patents, Washington, D.C. 20231 on

Date: *6/5/02*

Reg. No. 36,197

Signature:

[Signature]
Jong H. Lee

A claim to the Convention Priority Date pursuant to 35 U.S.C. § 119 of Application
No. 101 10 444.8 filed in the German Patent and Trademark Office on March 5, 2001 is hereby made.
To complete the claim to the Convention Priority Date, a certified copy of the priority application is
attached.

Dated: *6/5/02*

By:

[Signature]
Richard L. Mayer (Reg. No. 22,490)

KENYON & KENYON
One Broadway
New York, N.Y. 10004
(212) 425-7200 (telephone)
(212) 425-5288 (facsimile)

[Handwritten notes]
by
R. No.
36,197

BUNDESREPUBLIK DEUTSCHLAND



Prioritätsbescheinigung über die Einreichung einer Patentanmeldung

Aktenzeichen: 101 10 444.8

Anmeldetag: 05. März 2001

Anmelder/Inhaber: ROBERT BOSCH GMBH,
Stuttgart/DE

Bezeichnung: Verfahren und Vorrichtung zum Ermitteln
der Auslastung eines Rechengерäts

IPC: G 06 F 11/34

Die angehefteten Stücke sind eine richtige und genaue Wiedergabe der ursprünglichen Unterlagen dieser Patentanmeldung.

München, den 07. März 2002
Deutsches Patent- und Markenamt
Der Präsident
Im Auftrag

**CERTIFIED COPY OF
PRIORITY DOCUMENT**

Wallner

sind unterschiedliche Prioritäten zugeordnet. Die Tasks können sich gegenseitig unterbrechen. Das Steuerprogramm kann in einem kooperativen oder in einem preemptiven Modus abgearbeitet werden. Die vorliegende Erfindung gilt für die Abarbeitung von Tasks durch das Rechengerät im kooperativen und im preemptiven Modus gleichermaßen.

Die Abarbeitung einzelner Tasks eines Steuerprogramms im kooperativen Modus bedeutet, dass bei unterschiedlich priorisierten Tasks eine auszuführende höherpriorisierte Task zu einer Unterbrechung einer aktuell ausgeführten niederpriorisierten Task führt. Anders als im preemptiven Modus, bei dem eine auszuführende höherpriorisierte Task einen aktuell ausgeführten Prozess einer niederpriorisierten Task unterbricht, wartet im kooperativen Modus die höherpriorisierte Task das Ende des aktuell ausgeführten Prozesses der niederpriorisierten Task ab. Erst danach wird die niederpriorisierte Task unterbrochen und die höherpriorisierte Task ausgeführt. Wenn die höherpriorisierte Task fertig ist, wird die niederpriorisierte Task bei dem Prozess fortgesetzt, vor dem sie unterbrochen wurde.

Die Abarbeitung der Tasks eines Steuerprogramms im kooperativen Modus ist aus der DE 195 00 957 A1 bekannt. Die Unterbrechung einer niederpriorisierten Task durch eine höherpriorisierte Task gehört zu den Aufgaben eines Multi-Tasking Betriebssystems. Ein solches Multi-Tasking Betriebssystem, das sowohl den kooperativen Modus als auch den preemptiven Modus bei der Abarbeitung von Steuerprogrammen unterstützt, ist bspw. das Echtzeitbetriebssystem ERCOS^{EK} von der Firma ETAS Entwicklungs- und Applikationswerkzeuge für elektronische Systeme GmbH & Co. KG, Stuttgart, Deutschland (vgl. ETAS GmbH & Co. KG: ERCOS^{EK} V2.0.0 Manual, Stuttgart, 1998). Auf die DE 195 00 957 A1 und das ERCOS^{EK}-Handbuch wird

ausdrücklich Bezug genommen.

5 Während des Betriebs eines Rechengeräts ist dieses je nach
Anzahl der Prozessaufrufe und Abarbeitungsdauer der
aufgerufenen Prozesse mehr oder weniger stark ausgelastet.
Bei zeit- oder sicherheitskritischen Anwendungen, wie bspw.
der Steuerung von X-by-Wire-Anwendungen (Steer-by-Wire,
Brake-by-Wire, etc.) in einem Kraftfahrzeug, muss eine zu
starke Auslastung bzw. gar eine Überlastung des
10 Rechengeräts vermieden werden, da bei einer Überlastung
eine ordnungsgemäße Abarbeitung des Programms nicht mehr
gewährleistet werden kann. Um die Auslastung des
Rechengeräts zu überwachen und bei einer zu starken
Auslastung geeignete Gegenmassnahmen treffen zu können, ist
15 es bekannt, die Auslastung des Rechengeräts zu ermitteln.
Die Gegenmassnahmen bestehen bspw. darin, den Aufruf
weniger zeit- oder sicherheitskritische Prozesse gezielt zu
verzögern um den Aufruf und die Abarbeitung von besonders
zeit- oder sicherheitskritischen Prozessen zu ermöglichen.
20

Zum Ermitteln der Auslastung eines Rechengeräts sind
verschiedene Verfahren bekannt. Aus der DE 197 57 876 A1
ist es bspw. bekannt, immer dann, wenn das Rechengerät
nicht ausgelastet ist, d.h. kein Prozess auf dem
25 Rechengerät abgearbeitet wird, ein gesondertes
Leerlaufprogramm aufzurufen. Aus der Laufdauer oder der
Anzahl der Aufrufe des Leerlaufprogramms während eines
vorgebbaren Zeitintervalls kann die Auslastung des
Rechengeräts ermittelt werden ($\text{Auslastung}_{\text{Rechengerät}} [\%] =$
30 $100\% - \text{Laufdauer}_{\text{Leerlaufprogramm}} [\%]$). Die berechnete Auslastung
stellt einen über einen längeren Zeitintervall gefilterter
Mittelwert dar. Aus diesem Grund kann die Auslastung des
Rechengeräts mit dem bekannten Verfahren nur mit einer sehr
geringen Dynamik bestimmt werden.
35

Ein weiterer Nachteil des bekannten Verfahrens ist, dass

das Leerlaufprogramm bei einer starken Auslastung nur sehr selten aufgerufen wird und auf dem Rechenggerät abläuft. Wenn das betrachtete Zeitintervall zu kurz vorgegeben wird, kann es vorkommen, dass das Leerlaufprogramm in dem
5 Zeitintervall nicht aufgerufen wird und eine 100%-ige Auslastung ermittelt wird, obwohl die tatsächliche Auslastung geringer ist. Ebenso wird das Leerlaufprogramm bei einer schwachen Auslastung des Rechenggeräts nur sehr selten nicht aufgerufen, nämlich nur dann nicht, wenn
10 Prozesse des Computerprogramms abgearbeitet werden. Wenn das betrachtete Zeitintervall zu kurz vorgegeben wird, kann es vorkommen, dass das Leerlaufprogramm in dem Zeitintervall ständig abläuft und eine 0%-ige Auslastung ermittelt wird, obwohl die tatsächliche Auslastung höher
15 ist.

Der vorliegenden Erfindung liegt die Aufgabe zugrunde, einerseits die Auslastung eines Rechenggeräts möglichst genau und zuverlässig und mit einer hohen Dynamik und
20 andererseits die Abarbeitungsdauer von Prozessen oder Tasks eines auf dem Rechenggerät abgearbeiteten Computerprogramms unabhängig von Unterbrechungen durch Prozesse höherpriorer Tasks zu ermitteln.

25 Zur Lösung dieser Aufgabe schlägt die vorliegende Erfindung ausgehend von dem Verfahren der eingangs genannten Art vor, dass

- ein Zeitintervall derart gewählt wird, dass während des Zeitintervalls mindestens eine Task begonnen und wieder beendet wird;
30
- während des Zeitintervalls nach Beenden der oder jeder Task die Laufzeit der Task ermittelt wird; und
- falls die beendete Task von mindestens einer weiteren Task unterbrochen wurde, die Laufzeit der oder jeder weiteren Task von der ermittelten Laufzeit abgezogen wird.
35

Vorteile der Erfindung

5 Mit dem erfindungsgemäßen Verfahren können die unterbrechungsfreien Abarbeitungszeiten einer jeden in dem Recheng Gerät aktiven Task ermittelt werden. Die geschieht durch Einfügen geeigneter Programm-Befehle am Abarbeitungsbeginn und am Abarbeitungsende der Tasks des Computerprogramms. Durch eine Addition der Abarbeitungszeiten aller Tasks, die innerhalb des
10 vorgebbaren Zeitintervalls beendet werden, und eine anschließende Normierung auf das Zeitintervall, kann die prozentuale Auslastung des Recheng Geräts mit nahezu beliebiger Dynamik und ohne Filterung bzw. Mittelwertbildung ermittelt werden.

15 Gemäß einer vorteilhaften Weiterbildung der vorliegenden Erfindung wird vorgeschlagen, dass

- das Zeitintervall derart gewählt wird, dass während des Zeitintervalls mindestens zwei Tasks begonnen und
20 wieder beendet werden; und
- die Laufzeiten der beendeten Tasks in der Reihenfolge der Beendigung der Tasks ermittelt werden.

25 Gemäß einer bevorzugten Ausführungsform der vorliegenden Erfindung wird vorgeschlagen, dass

- zu Beginn des Verfahrens eine Variable (Unterbr) auf Null gesetzt wird;
- die Laufzeit der beendeten Task anhand der Gleichung
$$t_{\text{Laufzeit}} = t_{\text{Ende}} - t_{\text{Anfang}} - (\text{Unterbr}_{\text{Ende}} - \text{Unterbr}_{\text{Anfang}})$$

ermittelt wird; und
30
- ein neuer Wert für die Variable anhand der Gleichung
$$\text{Unterbr} = \text{Unterbr}_{\text{Anfang}} + (t_{\text{Ende}} - t_{\text{Anfang}})$$

ermittelt wird,

35 wobei t_{Anfang} der Wert eines während der Abarbeitung des Computerprogramms fortlaufenden Zeitzählers zu Beginn der Task, t_{Ende} der Wert des Zeitzählers nach Beenden der Task,

Unterbr_{Anfang} der Wert der Variablen zu Beginn der Task und Unterbr_{Ende} der Wert der Variablen nach Beenden der Task ist.

5 Vorteilhafterweise werden zum Ermitteln der Auslastung des Rechengeräts nach Ablauf des vorgegebenen Zeitintervalls die ermittelten Laufzeiten der Tasks addiert und in Verhältnis zu dem Zeitintervall gesetzt.

10 Vorzugsweise werden die ermittelten Laufzeiten der einzelnen Tasks jeweils in einer eigenen Speicherzelle des Rechengeräts, vorzugsweise in einer Random-Access-Memory (RAM)-Speicherzelle, abgelegt. Für eine nachfolgende
15 Berechnung der Auslastung muß gewährleistet werden, dass in den Speicherzellen aktuelle Laufzeitwerte zur Verfügung stehen. Das kann bspw. durch Löschen der Speicherzellen nach der Berechnung der Auslastung oder durch Überschreiben des Inhalts der Speicherzellen nach der Berechnung bspw. mit Null oder einem aktuellen Laufzeitwert für die
20 nachfolgende Berechnung.

Von besonderer Bedeutung ist die Realisierung des erfindungsgemäßen Verfahrens in der Form eines Speicherelements. Dabei ist auf dem Speicherelement ein
25 Computerprogramm gespeichert, das auf einem Rechengerät, insbesondere auf einem Mikroprozessor, ablauffähig und zur Ausführung des erfindungsgemäßen Verfahrens geeignet ist. In diesem Fall wird also die Erfindung durch ein auf dem Speicherelement abgespeichertes Computerprogramm
30 realisiert, so dass dieses mit dem Computerprogramm versehene Speicherelement in gleicher Weise die Erfindung darstellt wie das Verfahren, zu dessen Ausführung das Computerprogramm geeignet ist. Als Speicherelement kann insbesondere ein elektrisches Speichermedium zur Anwendung
35 kommen, bspw. ein Read-Only-Memory, ein Random-Access-Memory oder ein Flash-Memory.

Die Erfindung betrifft auch ein Computerprogramm, das zur Ausführung des erfindungsgemäßen Verfahrens geeignet ist, wenn es auf einem Recheng Gerät, insbesondere auf einem Mikroprozessor, abläuft. Besonders bevorzugt ist dabei, wenn das Computerprogramm auf einem Speicherelement, insbesondere auf einem Flash-Memory, abgespeichert ist.

Als eine weitere Lösung der Aufgabe der vorliegenden Erfindung wird ausgehend von der Vorrichtung zum Ermitteln der Auslastung eines Recheng Geräts der eingangs genannten Art vorgeschlagen, dass die Vorrichtung

- während eines vorgebbaren Zeitintervalls nach Beenden der oder jeder Task die Laufzeit der Task ermittelt; und
- falls die beendete Task von mindestens einer weiteren Task unterbrochen wurde, die Laufzeit der oder jeder weiteren Task von der ermittelten Laufzeit abzieht.

Die erfindungsgemäße Vorrichtung ist bspw. als ein Steuergerät eines Kraftfahrzeugs ausgebildet, das zur Steuerung/Regelung von technischen Vorgängen und anderer Funktionen in dem Kraftfahrzeug dient. Das Steuergerät weist ein Recheng Gerät, insbesondere einen Mikroprozessor, auf, auf dem ein Steuerprogramm ablauffähig ist. Das Steuerprogramm ist in mehrere Tasks unterteilt und jede Task umfasst mindestens einen Prozess. In der Vorrichtung sind Mittel zur Ausführung des erfindungsgemäßen Verfahrens realisiert. Dadurch können die unterbrechungsfreien Abarbeitungszeiten einer jeden in dem Recheng Gerät aktiven Task ermittelt werden. Durch eine Addition der Abarbeitungszeiten aller Tasks, die innerhalb des vorgebbaren Zeitintervalls beendet werden, und eine anschließende Normierung auf das Zeitintervall, kann zudem die prozentuale Auslastung des Recheng Geräts ermittelt werden.

Zeichnungen

Weitere Merkmale, Anwendungsmöglichkeiten und Vorteile der Erfindung ergeben sich aus der nachfolgenden Beschreibung von Ausführungsbeispielen der Erfindung, die in der
5 Zeichnung dargestellt sind. Dabei bilden alle beschriebenen oder dargestellten Merkmale für sich oder in beliebiger Kombination den Gegenstand der Erfindung, unabhängig von ihrer Zusammenfassung in den Patentansprüchen oder deren
10 Rückbeziehung sowie unabhängig von ihrer Formulierung bzw. Darstellung in der Beschreibung bzw. in der Zeichnung. Es zeigen:

Figur 1 ein Ablaufdiagramm eines in drei Tasks A, B, C
15 unterteilten Steuerprogramms;

Figur 2 ein Ablaufdiagramm eines erfindungsgemäßen
Verfahrens gemäß einer bevorzugten
Ausführungsform; und

20 Figur 3 eine erfindungsgemäße Vorrichtung zur Abarbeitung des Verfahrens aus Figur 2 gemäß einer bevorzugten Ausführungsform.

25 Beschreibung der Ausführungsbeispiele

In Figur 3 ist ein Steuergerät eines Kraftfahrzeugs in seiner Gesamtheit mit dem Bezugszeichen 1 bezeichnet. Das Steuergerät 1 weist ein Rechenggerät auf, das als ein
30 Mikroprozessor 2 ausgebildet ist. Auf einem Speicherelement des Steuergeräts 1, das als ein Flash-Memory 3 ausgebildet ist, ist ein Computerprogramm gespeichert. Das Computerprogramm ist als ein Steuerprogramm zur Steuerung von technischen Vorgängen insbesondere in einem
35 Kraftfahrzeug ausgebildet. Das Steuerprogramm kann auf dem Mikroprozessor 2 abgearbeitet werden. Zur Datenübertragung

ist zwischen dem Mikroprozessor 2 und dem Flash-Memory 3 eine Datenverbindung 4, die bspw. als eine Bus-Leitung ausgebildet ist, vorgesehen. An dem Steuergerät 1 liegen verschiedene Eingangsgrößen 5 an, die bspw. die Signale von Sensoren und Messwertaufnehmern sind. Die Eingangsgrößen 5 charakterisieren den Zustand der zu steuernden Funktionen, den Zustand des Kraftfahrzeugs oder andere Zustände wie bspw. die Witterungsverhältnisse. Anhand der Eingangsgrößen 5 werden in dem Steuergerät 1 Ausgangsgrößen 6 ermittelt, die zur Ansteuerung von Aktoren oder Stellern dienen. Die Ausgangsgrößen 6 sind bspw. Sollwerte für Steuerungen oder Regelungen.

Das Steuerprogramm ist in mehrere Tasks A, B, C unterteilt (vgl. Figur 1), wobei jede Task wiederum mindestens einen Prozess umfasst. Eine Task A, B, C wird zu einem bestimmten Zeitpunkt oder regelmäßig mit einer bestimmten Abtastzeit aufgerufen und kann in einem kooperativen oder in einem preemptiven Modus abgearbeitet werden. Jeder Task A, B, C ist eine bestimmte Priorität zugeordnet. Die Task A hat die niedrigste Priorität, die Task C die höchste. Wenn während der Abarbeitung des Steuerprogramms zwei Tasks gleichzeitig ausgeführt werden sollen, werden die Prioritäten der beiden Tasks verglichen und die Task mit der höheren Priorität als erste abgearbeitet.

Wenn bspw. die Task A abgearbeitet wird und die Task B ausgeführt werden soll, können je nach der von einem Programmierer gewählten Konfiguration der Tasks verschiedene Fälle auftreten:

Falls die Task A eine höhere Priorität als die Task B hat (in Figur 1 nicht der Fall), wird mit der Ausführung der Task B gewartet bis die Task A beendet ist.

Falls die Task B eine höhere Priorität als die Task A hat

(Beispiel aus Figur 1), wird die Abarbeitung der Task A unterbrochen und die Task B ausgeführt. Falls der Programmierer die Ausführung der Tasks in dem kooperativen Modus gewählt hat, wird mit der Ausführung der Task B auf das Ende des aktuellen Prozesses der Task A gewartet. Sobald dieser Prozess beendet ist, wird die Task A unterbrochen und die Task B ausgeführt. Wenn die Task B beendet ist, wird die Task A zu Beginn des Prozesses, vor dem Sie zur Ausführung der Task B unterbrochen wurde, weiter abgearbeitet.

Falls der Programmierer die Abarbeitung der Tasks in dem preemptiven Modus gewählt hat, unterbricht die Task B den aktuellen Prozess der Task A und die Task B wird unmittelbar ausgeführt. Anschließend wird die Task A bei dem unterbrochenen Prozess weiter abgearbeitet.

Die Unterbrechung einer Task durch eine andere Task mit einer höheren Priorität gehört zu den Aufgaben eines Multi-Tasking Betriebssystems. Die Laufzeit der Prozesse schwankt je nach der Belastung des Rechengenüts, auf dem das Steuerprogramm abgearbeitet wird. Aus diesem Grund und aufgrund der möglichen, von anderen Tasks verursachten Unterbrechungen kann die Reihenfolge der Prozessaufrufe bei einer mehrmaligen Ausführung ein und desselben Steuerprogramms unterschiedlich sein. Die Laufzeiten der einzelnen Tasks innerhalb eines vorgebbaren Zeitintervalls kann also Schwankungen unterworfen sein.

Um die Abarbeitungsdauer von Prozessen oder Tasks A, B, C eines auf dem Mikroprozessor 2 abgearbeiteten Steuerprogramms unabhängig von Unterbrechungen durch Prozesse höherpriorer Tasks zu ermitteln, wird ein erfindungsgemäßes Verfahren vorgeschlagen, von dem ein Ablaufdiagramm in Figur 2 dargestellt ist. Anhand der Abarbeitungsdauer der einzelnen Prozesse oder Tasks A, B, C

innerhalb eines vorgebbaren Zeitintervalls T kann anhand des erfindungsgemäßen Verfahrens auch die Auslastung des Mikroprozessors 2 genau und zuverlässig und mit einer hohen Dynamik ermittelt werden.

5

10

15

Das Verfahren beginnt in einem Funktionsblock 10. In einem Funktionsblock 11 wird das Zeitintervall T vorgegeben. Das Zeitintervall T wird derart gewählt, dass während des Zeitintervalls T mindestens eine Task A, B, C begonnen und auch wieder beendet wird. In einem Funktionsblock 12 wird die Abarbeitung des Steuerprogramms (vgl. Figur 2) auf dem Mikroprozessor 2 ganz normal eingeleitet. In einem Abfrageblock 13 wird überprüft, ob eine Task beendet wurde. Falls ja, wird in einem Funktionsblock 14 die Abarbeitungsdauer $t_{\text{Laufzeit_Task}}$ der beendeten Task nach folgender Gleichung berechnet:

$$t_{\text{Laufzeit_Task}} = t_{\text{Ende}} - t_{\text{Anfang}} - (\text{Unterbr}_{\text{Ende}} - \text{Unterbr}_{\text{Anfang}}),$$

20

25

wobei t_{Anfang} der Wert eines während der Abarbeitung des Computerprogramms fortlaufenden Zeitzählers zu Beginn der Task, t_{Ende} der Wert des Zeitzählers nach Beenden der Task, $\text{Unterbr}_{\text{Anfang}}$ der Wert einer Variablen Unterbr zu Beginn der Task und $\text{Unterbr}_{\text{Ende}}$ der Wert der Variablen Unterbr nach Beenden der Task ist.

Anschließend wird in einem Funktionsblock 15 ein neuer Wert für die Variable Unterbr nach folgender Gleichung berechnet:

30

$$\text{Unterbr} = \text{Unterbr}_{\text{Anfang}} + (t_{\text{Ende}} - t_{\text{Anfang}}).$$

35

In einem Abfrageblock 16 wird überprüft, ob das vorgegebene Zeitintervall T bereits abgelaufen ist oder nicht. Falls ja wird in einem Funktionsblock 17 die Summe t_{ges} der Abarbeitungsdauer $t_{\text{Laufzeit_Task}}$ aller Tasks, die in dem

vorgegebenen Zeitintervall T beendet wurden, gebildet. In einem Funktionsblock 18 wird die Gesamt-Abarbeitungsdauer t_{ges} dann zur Ermittlung der Auslastung des Mikroprozessors 2 in Verhältnis zu dem vorgegebenen Zeitintervall T gesetzt. In einem Funktionsblock 19 ist das erfindungsgemäße Verfahren beendet.

Falls sich in dem Abfrageblock 13 ergibt, dass keine Task beendet ist, oder falls sich in dem Abfrageblock 16 ergibt, dass das vorgegebene Zeitintervall T noch nicht abgelaufen ist, wird zu dem Funktionsblock 12 verzweigt, wo die Abarbeitung des Steuerprogramms fortgesetzt wird.

Nachfolgend wird das erfindungsgemäße Verfahren aus Figur 2 anhand des Steuerprogramms aus Figur 1 beispielhaft erläutert. Das Zeitintervall T wird auf $T = 610 - 100 = 510$ Millisekunden gesetzt. Die Variable Unterbr wird auf Null gesetzt. Anschließend wird der Ablauf des Steuerprogramms begonnen. Die erste Task die beendet wird, ist die Task C. Die Abarbeitungsdauer dieser Task C wird ermittelt:

$$\begin{aligned} t_{\text{Laufzeit}_C} &= t_{\text{Ende}} - t_{\text{Anfang}} - (\text{Unterbr}_{\text{Ende}} - \text{Unterbr}_{\text{Anfang}}) \\ &= 270 - 210 - 0 \\ &= 60 \text{ Millisekunden.} \end{aligned}$$

Die Differenz $(\text{Unterbr}_{\text{Ende}} - \text{Unterbr}_{\text{Anfang}})$ wird auf Null gesetzt, da die Task C nicht unterbrochen wurde. Anschließend wird die Variable Unterbr ermittelt:

$$\begin{aligned} \text{Unterbr} &= \text{Unterbr}_{\text{Anfang}} + (t_{\text{Ende}} - t_{\text{Anfang}}) \\ &= 0 + (270 - 210) \\ &= 60 \text{ Millisekunden.} \end{aligned}$$

Die nächste Task, die beendet wird, ist wiederum die Task C. Für diese Task C wird die Abarbeitungsdauer t_{Laufzeit_C} und der neue Wert für die Variable Unterbr berechnet:

$$\begin{aligned} t_{\text{Laufzeit}_C} &= t_{\text{Ende}} - t_{\text{Anfang}} - (\text{Unterbr}_{\text{Ende}} - \text{Unterbr}_{\text{Anfang}}) \\ &= 350 - 310 - 0 \\ &= 40 \text{ Millisekunden,} \end{aligned}$$

5

$$\begin{aligned} \text{Unterbr} &= \text{Unterbr}_{\text{Anfang}} + (t_{\text{Ende}} - t_{\text{Anfang}}) \\ &= 60 + (350 - 310) \\ &= 100 \text{ Millisekunden.} \end{aligned}$$

10 Als nächste Task ist die Task B beendet. Für die Task B wird die Abarbeitungsdauer t_{Laufzeit_B} und der neue Wert für die Variable Unterbr berechnet:

15

$$\begin{aligned} t_{\text{Laufzeit}_B} &= t_{\text{Ende}} - t_{\text{Anfang}} - (\text{Unterbr}_{\text{Ende}} - \text{Unterbr}_{\text{Anfang}}) \\ &= 420 - 150 - (100 - 0) \\ &= 170 \text{ Millisekunden,} \end{aligned}$$

20

$$\begin{aligned} \text{Unterbr} &= \text{Unterbr}_{\text{Anfang}} + (t_{\text{Ende}} - t_{\text{Anfang}}) \\ &= 0 + (420 - 150) \\ &= 270 \text{ Millisekunden.} \end{aligned}$$

Schließlich ist die Task A beendet. Für die Task A wird die Abarbeitungsdauer t_{Laufzeit_A} und der neue Wert für die Variable Unterbr berechnet:

25

$$\begin{aligned} t_{\text{Laufzeit}_A} &= t_{\text{Ende}} - t_{\text{Anfang}} - (\text{Unterbr}_{\text{Ende}} - \text{Unterbr}_{\text{Anfang}}) \\ &= 530 - 100 - (270 - 0) \\ &= 160 \text{ Millisekunden,} \end{aligned}$$

30

$$\begin{aligned} \text{Unterbr} &= \text{Unterbr}_{\text{Anfang}} + (t_{\text{Ende}} - t_{\text{Anfang}}) \\ &= 0 + (530 - 100) \\ &= 430 \text{ Millisekunden.} \end{aligned}$$

35 Die Summe t_{ges} der Abarbeitungsdauer der einzelnen Tasks A, B, C ist dann $t_{\text{ges}} = 430$ Millisekunden. Bei einem vorgegebenen Zeitintervall T von 510 Millisekunden ergibt sich mit dem erfindungsgemäßen Verfahren somit eine

$$t_{\text{Laufzeit_Task}} = t_{\text{Ende}} - \text{Unterbr}_{\text{Ende}} + \text{Hilfsvar}$$

$$\text{Unterbr} = t_{\text{Ende}} + \text{Hilfsvar}.$$

5 Gemäß noch einer anderen Berechnung der Auslastung des Rechengegeräts werden lediglich zwei Hilfsvariable Hilfsvar1 und Hilfsvar2 benötigt und es müssen nur drei Additionen ausgeführt werden. Am Anfang der Task wird die erste Hilfsvariable Hilfsvar1 berechnet:

10

$$\text{Hilfsvar1} = \text{Unterbr}_{\text{Anfang}} - t_{\text{Anfang}}$$

Am Ende der Task wird die zweite Hilfsvariable Hilfsvar2, die Laufzeit $t_{\text{Laufzeit_Task}}$ und die Variable Unterbr berechnet:

15

$$\text{Hilfsvar2} = t_{\text{Ende}} + \text{Hilfsvar1}$$

$$t_{\text{Laufzeit_Task}} = \text{Hilfsvar2} - \text{Unterbr}_{\text{Ende}}$$

20

$$\text{Unterbr} = \text{Hilfsvar2}.$$

5 01.03.2001
Robert Bosch GmbH, 70469 Stuttgart

Ansprüche

- 10 1. Verfahren zum Ermitteln der Auslastung eines
Rechengeräts (2), auf dem ein Computerprogramm abgearbeitet
wird, wobei das Computerprogramm in mehrere Tasks (A, B, C)
unterteilt ist und jede Task (A, B, C) mindestens einen
Prozess umfasst, **dadurch gekennzeichnet**, dass
- 15 - ein Zeitintervall (T) derart gewählt wird, dass
während des Zeitintervalls (T) mindestens eine Task
(A, B, C) begonnen und wieder beendet wird;
- während des Zeitintervalls (T) nach Beenden der oder
jeder Task (A, B, C) die Laufzeit (t_{Laufzeit}) der Task
20 (A, B, C) ermittelt wird; und
- falls die beendete Task (A, B, C) von mindestens einer
weiteren Task (A, B, C) unterbrochen wurde, die
Laufzeit der oder jeder weiteren Task (A, B, C) von
der ermittelten Laufzeit (t_{Laufzeit}) abgezogen wird.
- 25 2. Verfahren nach Anspruch 1, dadurch gekennzeichnet,
dass
- 30 - das Zeitintervall (T) derart gewählt wird, dass
während des Zeitintervalls (T) mindestens zwei Tasks
(A, B, C) begonnen und wieder beendet werden; und
- die Laufzeiten (t_{Laufzeit}) der beendeten Tasks (A, B, C)
in der Reihenfolge der Beendigung der Tasks (A, B, C)
ermittelt werden.
- 35 3. Verfahren nach Anspruch 2, dadurch gekennzeichnet,
dass

- zu Beginn des Verfahrens eine Variable (Unterbr) auf Null gesetzt wird;
- die Laufzeit (t_{Laufzeit}) der beendeten Task (A, B, C) anhand der Gleichung

$$t_{\text{Laufzeit}} = t_{\text{Ende}} - t_{\text{Anfang}} - (\text{Unterbr}_{\text{Ende}} - \text{Unterbr}_{\text{Anfang}})$$

ermittelt wird; und

- ein neuer Wert für die Variable (Unterbr) anhand der Gleichung

$$\text{Unterbr} = \text{Unterbr}_{\text{Anfang}} + (t_{\text{Ende}} - t_{\text{Anfang}})$$

ermittelt wird,

wobei t_{Anfang} der Wert eines während der Abarbeitung des Computerprogramms fortlaufenden Zeitzählers zu Beginn der Task (A, B, C), t_{Ende} der Wert des Zeitzählers nach Beenden der Task (A, B, C), $\text{Unterbr}_{\text{Anfang}}$ der Wert der Variablen (Unterbr) zu Beginn der Task (A, B, C) und $\text{Unterbr}_{\text{Ende}}$ der Wert der Variablen (Unterbr) nach Beenden der Task (A, B, C) ist.

4. Verfahren nach Anspruch 2 oder 3, dadurch gekennzeichnet, dass die ermittelten Laufzeiten (t_{Laufzeit}) der einzelnen Tasks (A, B, C) jeweils in einer eigenen Speicherzelle des Rechengeräts (2), vorzugsweise in einer Random-Access-Memory (RAM)-Speicherzelle, abgelegt werden.

5. Verfahren nach einem der Ansprüche 2 bis 4, dadurch gekennzeichnet, dass zum Ermitteln der Auslastung des Rechengeräts (2) nach Ablauf des vorgegebenen Zeitintervalls (T) die ermittelten Laufzeiten (t_{Laufzeit}) der Tasks (A, B, C) addiert und in Verhältnis zu dem Zeitintervall (T) gesetzt werden.

6. Speicherelement, insbesondere Read-Only-Memory, Random-Access-Memory oder Flash-Memory (3), auf dem ein Computerprogramm gespeichert ist, das auf einem Rechengerät (2), insbesondere auf einem Mikroprozessor, ablauffähig und zur Ausführung eines Verfahrens nach einem der Ansprüche 1

bis 5 geeignet ist.

5 7. Computerprogramm, dadurch gekennzeichnet, dass das Computerprogramm zur Ausführung eines Verfahrens nach einem der Ansprüche 1 bis 5 geeignet ist, wenn es auf einem Recheng Gerät (2), insbesondere auf einem Mikroprozessor, abläuft.

10 8. Computerprogramm nach Anspruch 7, dadurch gekennzeichnet, dass das Computerprogramm auf einem Speicherelement, insbesondere auf einem Flash-Memory (3), abgespeichert ist.

15 9. Vorrichtung (1) zum Ermitteln der Auslastung eines Rechengeräts (2), auf dem ein Computerprogramm abarbeitbar ist, wobei das Computerprogramm in mehrere Tasks (A, B, C) unterteilt ist und jede Task (A, B, C) mindestens einen Prozess umfasst, dadurch gekennzeichnet, dass die Vorrichtung

- 20 - während eines vorgebbaren Zeitintervalls (T) nach Beenden der oder jeder Task (A, B, C) die Laufzeit (t_{Laufzeit}) der Task (A, B, C) ermittelt; und
- 25 - falls die beendete Task (A, B, C) von mindestens einer weiteren Task (A, B, C) unterbrochen wurde, die Laufzeit der oder jeder weiteren Task (A, B, C) von der ermittelten Laufzeit (t_{Laufzeit}) abzieht.

30 10. Vorrichtung (1) nach Anspruch 9, dadurch gekennzeichnet, dass die Vorrichtung Mittel zur Ausführung eines Verfahrens nach einem der Ansprüche 2 bis 5 aufweist.

5 01.03.2001
Robert Bosch GmbH, 70469 Stuttgart

10 Verfahren und Vorrichtung zum Ermitteln der Auslastung
eines Rechengeräts

Zusammenfassung

15 Die Erfindung betrifft ein Verfahren und eine Vorrichtung
(1) zum Ermitteln der Auslastung eines Rechengeräts (2).
Auf dem Rechengerät (2) wird ein Computerprogramm
abgearbeitet. Das Computerprogramm ist in mehrere Tasks (A,
B, C) unterteilt und jede Task (A, B, C) umfasst mindestens
einen Prozess. Um einerseits die Auslastung des
20 Rechengeräts (2) möglichst genau und zuverlässig und mit
einer hohen Dynamik und andererseits die Abarbeitungsdauer
der Prozesse oder Tasks (A, B, C) des Computerprogramms
unabhängig von Unterbrechungen durch Prozesse höherpriorer
Tasks ermitteln zu können, wird vorgeschlagen, dass
25 - ein Zeitintervall (T) derart gewählt wird, dass
während des Zeitintervalls (T) mindestens eine Task
(A, B, C) begonnen und wieder beendet wird;
- während des Zeitintervalls (T) nach Beenden der oder
jeder Task (A, B, C) die Laufzeit (t_{Laufzeit}) der Task
30 (A, B, C) ermittelt wird; und
- falls die beendete Task (A, B, C) von mindestens einer
weiteren Task (A, B, C) unterbrochen wurde, die
Laufzeit der oder jeder weiteren Task (A, B, C) von
der ermittelten Laufzeit (t_{Laufzeit}) abgezogen wird.
35 (Figur 2)

2 / 2

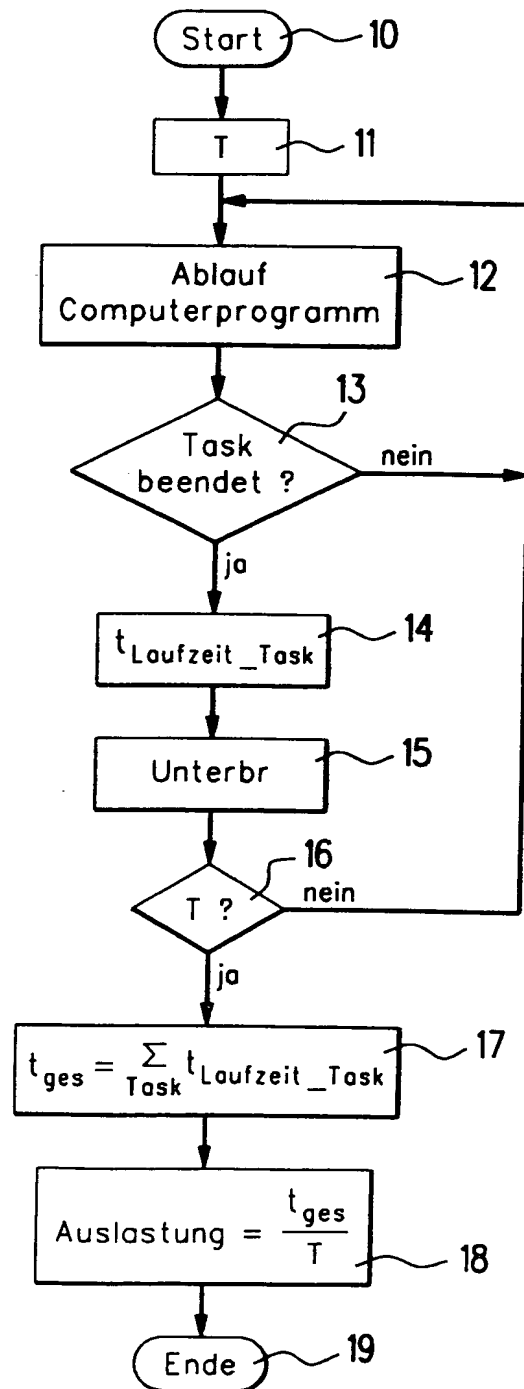


Fig. 2

1 / 2

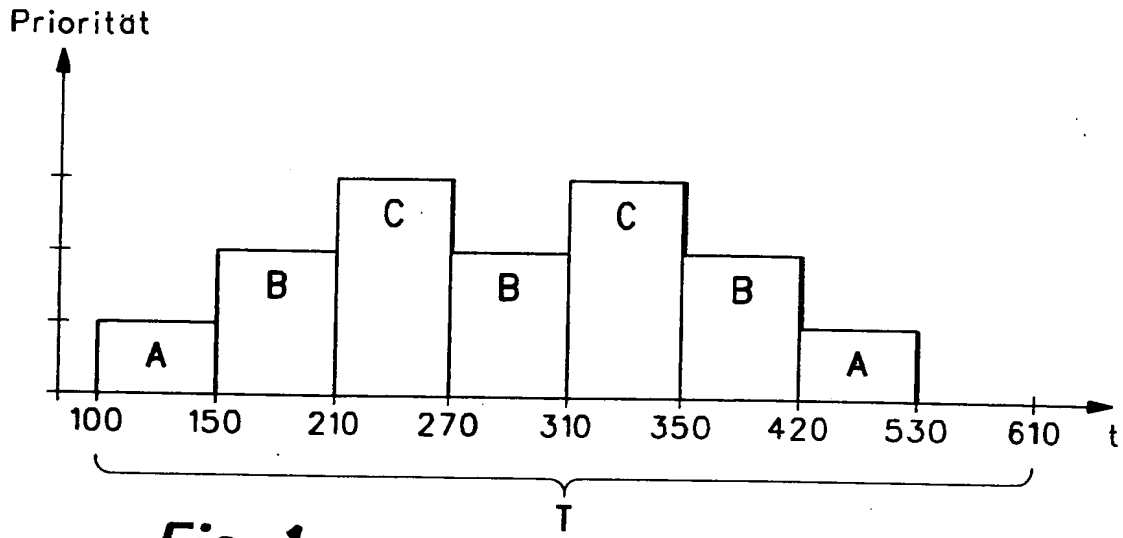


Fig. 1

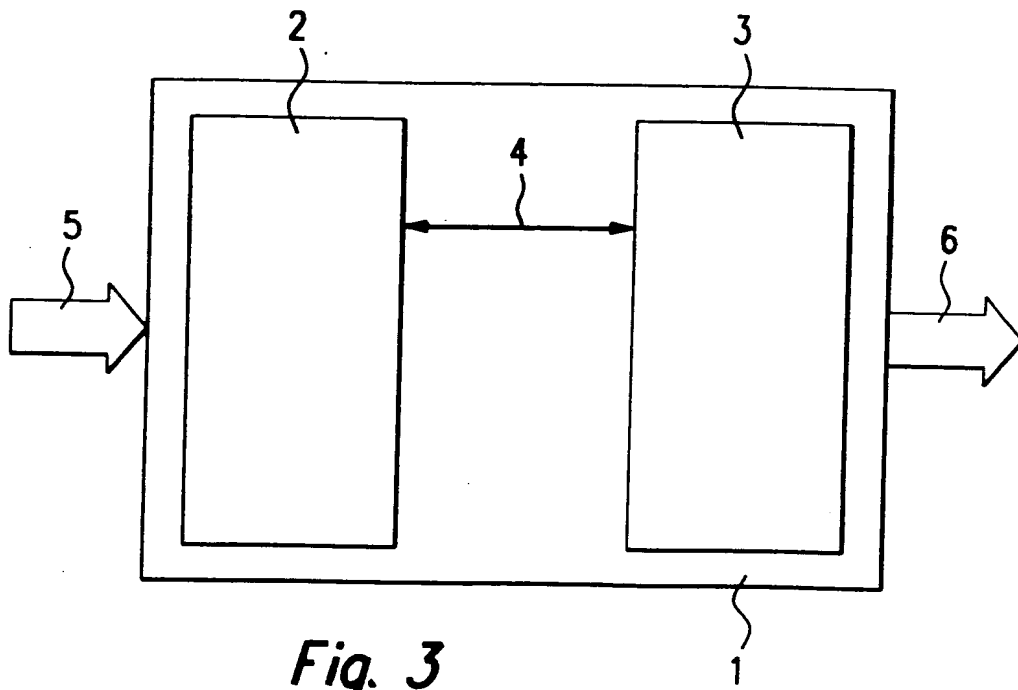


Fig. 3

2 / 2

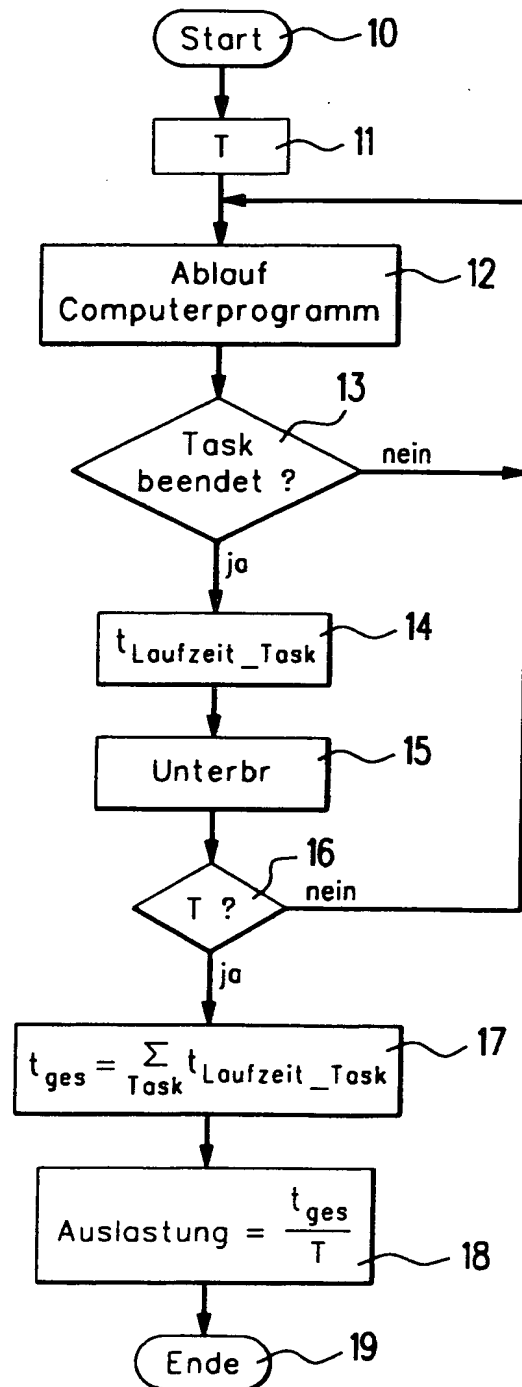


Fig. 2



Creation date: 10-18-2004
Indexing Officer: TNGUYEN64 - TUAN NGUYEN
Team: OIPEBackFileIndexing
Dossier: 10092225

Legal Date: 07-19-2002

No.	Doccode	Number of pages
1	IDS	2
2	NPL	33

Total number of pages: 35

Remarks:

Order of re-scan issued on